H8 Simple Simulator

# H8SS

H8/3664F 簡易シミュレータ

# 取扱説明書

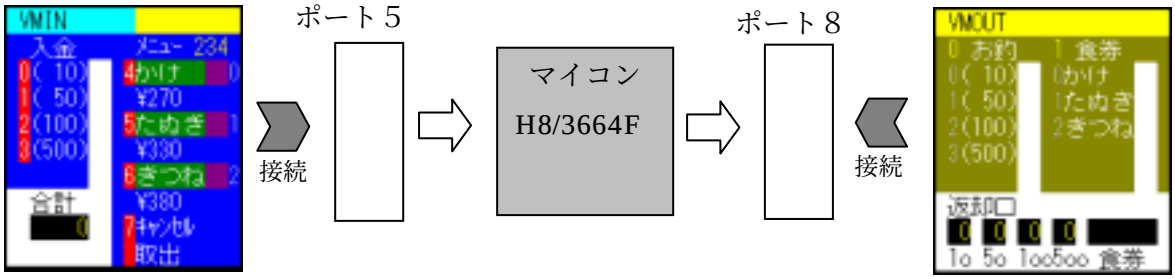
## Vendering Machine

## 追加説明

## １．Ｈ８ＳＳ　拡張機能概要

　Ｈ８ＳＳに自動販売機の簡易的な入力部・出力部を付け加えたものです。 ※ＶＭ（Vendering Machine）

　ユーザインターフェース部のプログラムをする必要が無いので、内部プログラミングに集中することが可能です。




## ２．使用可能ポートと利用可能ディバイス

| ポート | 用途 | 利用可能ディバイス |
|---|---|---|
| ポート５ | ８ビット入力専用ポート | **VMIN, SWITCH** |
| ポート８ | ８ビット出力専用ポート | **VMOUT, LED, LED7, SEG7, SEG7D, STEPMT, DCMT** |
| ポートＢ | ４チャンネル<br>アナログ入力専用ポート | **ANALOG(VMIN,VMOUT と排他利用)** |

## ３．プログラムへの組み込みにおける注意（後で詳細に説明します）

（１） Ｈ８ＳＳへのデータの受け渡し処理が随所で必要です。
　　データレジスタとＨ８ＳＳとのデータ受け渡しを、指定のタイミングで行う必要があります。

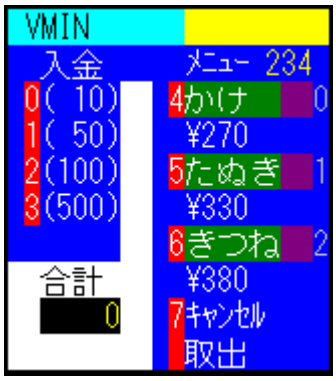
（２） 実行後に、各ポートの接続ディバイスを変更することができますが、
　　明示的にＨ８ＳＳの初期化をすることもできます。

（３） 実際には行うべき細部の手続きは省略されています。

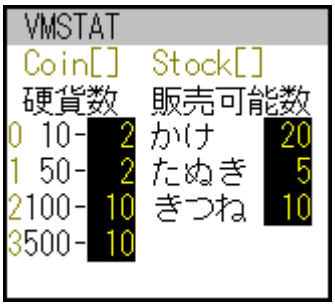
```
#include "3664.h"
void main(void)
{
   InitH8SS(NONE,LED,NONE);  （２）任意
    IO.PDR8.BYTE=0x55; io;  （１）
    while(1);
}
```


©2011 uaubn

## ４．接続可能ディバイス一覧

| VMIN（自動販売機操作部） | VMOUT（自動販売機出力部） |
|---|---|
|  | |

**VMSTAT（内部状況、VMIN,VMOUT 使用で自動表示）**



## ５．Ｈ８ＳＳでのプログラム開発の方法

（１）Ｈ８ＳＳでの各ポートへ接続するディバイスを InitH8SS() 関数でも設定できます。

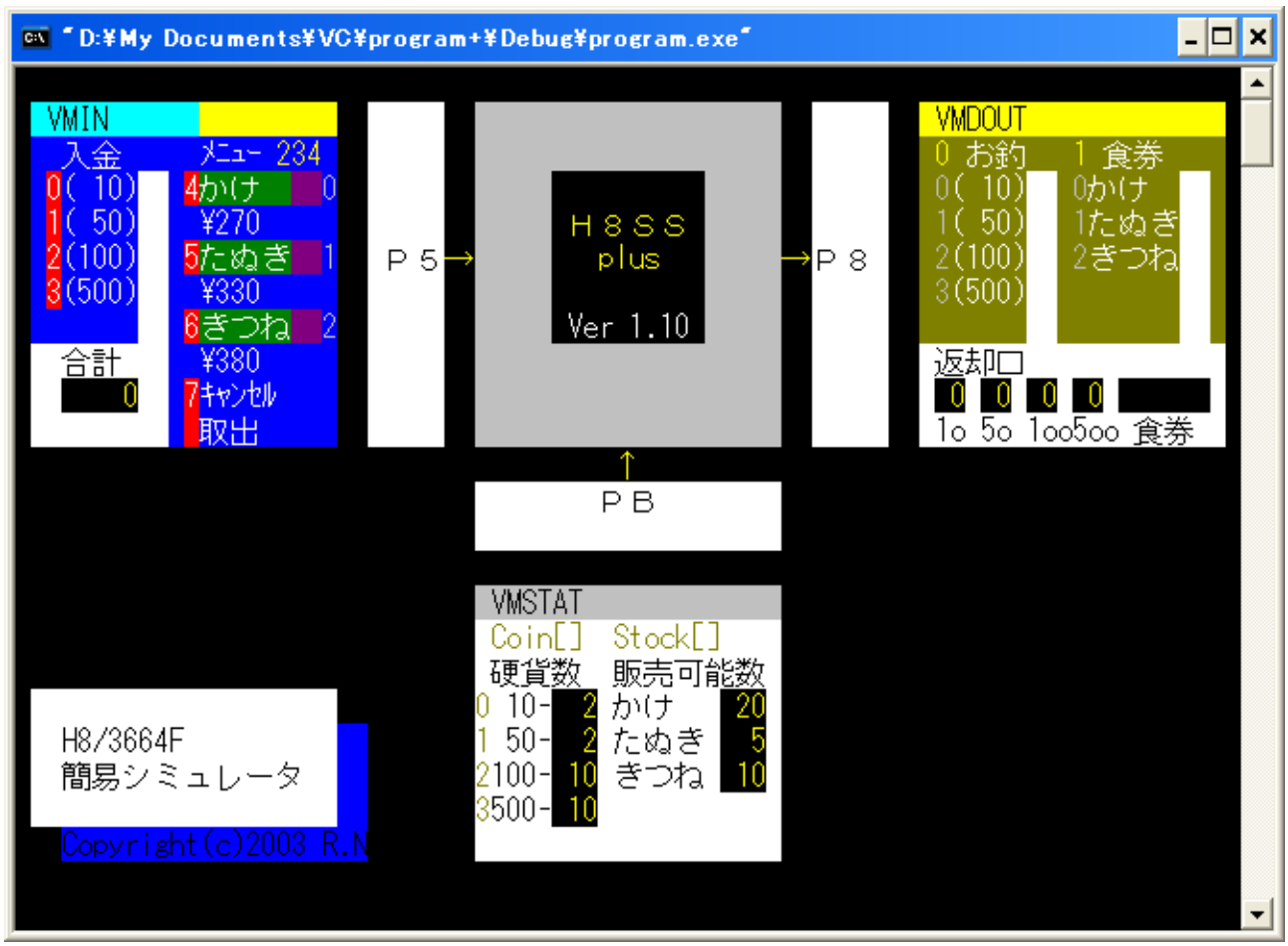
関数仕様

| | |
|---|---|
| 形式 | **Void InitH8SS(int P5Device, int P8Device, int PBDevice)** |
| 機能 | Ｈ８ＳＳでの各ポートへ接続するディバイスを設定します。 |
| 引数 | **P5Device:** ポート５に接続するディバイス指定<br>**P8Device:** ポート８に接続するディバイス指定<br>**PBDevice:** ポートＢに接続するディバイス指定 |
| 指定ディバイス | （ポート５用）VMIN, **SWITCH**<br>（ポート８用）VMOUT, **LED, LED7, SEG7, SEG7D, STEPMT, DCMT, LCD**<br>（ポートＢ用）**ANALOG(VMIN, VMOUT とは排他制御)**<br>（各ポート兼用）**NONE：**接続無し、 |
| 使用例 | **InitH8SS(SWITCH,SEG7,NONE);**<br>スイッチと７セグメントを接続<br>**InitH8SS(NONE,STEPMT,NONE);**<br>ステッピングモータのみ接続<br>**InitH8SS(NONE,LCD,ANALOG);**<br>アナログ入力とＬＣＤを接続 |


©2011 uaubn

## ６．Ｈ８ＳＳ　(Vendering Machine)　操作方法

(1) 実行画面




(2) キー操作

| 対象ディバイス | 操作キー | 動作 |
|---|---|---|
| **ＶＭＩＮ** | 0 ～ 7 | 自動販売機の入力操作を行います。<br>(0:B0, 1:B1, ・・・, 7:B7) |
| **ＶＭＩＮ**<br>**ＶＭＯＵＴ** | スペース | **投入金額合計値をクリアします。**<br>**返却口の内容を取り出します。** |
| **Ｈ８ＳＳ** | Q | Ｈ８ＳＳ(Vendering Machine)の終了 |


©2011 uaubn

## ７．各ディバイスの特徴

### 自動販売機操作部(VMIN)

ポート５の、キー入力された番号ビットが１に、その他のビットは０になります。
何もキーが押されていない場合は、全部のビットは０になっています。

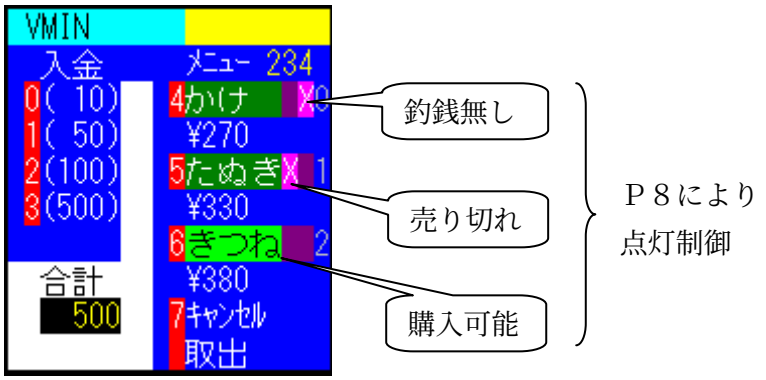
3 ５００円投入
↓

| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

6 きつねそば購入
↓

| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |

ＶＭＩＮは、ポート８の出力により、以下のように、「購入可能」「売り切れ」「釣銭無し」ランプを点灯させることができます。




入力方法例

| 項目 | キー入力操作 | 備考 |
|---|---|---|
| お金の入力 | 0 ～ 3 の入力で、それぞれの金額が入力されることになります。<br>0:１０円　1:５０円　2:１００円　3:５００円 | 合計表示のみは H8SS+1 内部機能 |
| 購入 | 購入可能ランプが点灯した番号（ 4 ～ 6 ）を入力すると、食券が発券され、おつりが出てきます。 | |
| キャンセル | お金を入れた状態で、7 を入力すると、キャンセルされ、投入金額が返却されます。 | |
| 返却口よりの取出 | スペースでおつりや、食券を取り出します。 | H8SS+1 内部機能 |


©2011 uaubn

## 自動販売機出力部(VMOUT)

ポート８に信号を出力することで、食券発券動作や、お釣りの返却動作などを行います。

H8SS+1 の内部機能で、返却・発券動作により、返却口にでたものを表示します。

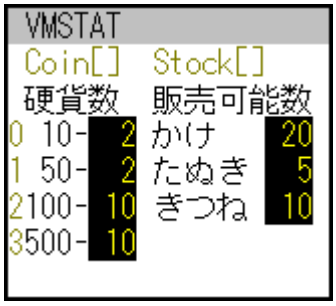
操作方法

| | Ｐ８出力データ | | | 備考 |
|---|---|---|---|---|
| | 上位 4bit | 下位 4bit | 内容 | |
| お釣り返却 | **0000** | **0000 → 0001** | 10 円１枚返却 | 下位 4bit 中の複数のビットが１の場合､最下位の１のビット操作のみが行われます。 |
| | | **0000 → 0010** | 50 円１枚返却 | |
| | | **0000 → 0100** | 100 円１枚返却 | |
| | | **0000 → 1000** | 500 円１枚返却 | |
| 食券発券 | **0001** | **0000 → 0001** | 食券「かけ」発券 | |
| | | **0000 → 0010** | 食券「たぬき」発券 | |
| | | **0000 → 0100** | 食券「きつね」発券 | |
| 購入可能ランプ | **0010** | **---?** | 「かけ」 | 一度出力すると､次の出力があるまで各ランプの状態保持されます。<br><br>? bit=0 ランプ消灯<br>? bit=1 ランプ点灯 |
| | | **--?-** | 「たぬき」 | |
| | | **-?--** | 「きつね」 | |
| 売り切れランプ | **0011** | **---?** | 「かけ」 | |
| | | **--?-** | 「たぬき」 | |
| | | **-?--** | 「きつね」 | |
| 釣銭切れランプ | **0100** | **---?** | 「かけ」 | |
| | | **--?-** | 「たぬき」 | |
| | | **-?--** | 「きつね」 | |


©2011 uaubn

**内部状況(VMSTAT)**

Ｈ８ＳＳ＋１の内部機能で、自動販売機内部の硬貨保持数と、販売可能数を表示しています。
公開変数の Coin[]と Stock[]をプログラム処理中で変更すると、その値が表示されます。




**公開配列変数**

次の変数が公開されています。プログラム開始時に値を変更することで、それぞれの内容を変更することができます。

| | 配列変数名 | 内容 |
|---|---|---|
| 内部保持硬貨数 | int Coin[4] | [0]10 円、[1]50 円、[2]100 円、[3]500 円の硬貨数 |
| 販売可能食券数 | int Stock[3] | [0]１つめ、[1]２つめ、[2]３つめの販売可能食券数 |
| メニュー項目 | char *Menu[3] | [0]１つめ、[1]２つめ、[2]３つめのメニュー表記 |
| 価格 | int Price[3] | [0]１つめ、[1]２つめ、[2]３つめの価格 |

# 「食券自動販売機」仕様書

| | |
|---|---|
| プログラム名 | hanbaiki.c |
| プログラム概要 | 食券の自動販売機の動作を模擬的に行うプログラムの作成<br>対象は「おそば屋さん」でメニューは３つ、使用可能硬貨は４種類に限定する |
| プログラム詳細 | １．VMIN 画面に現在の内部の状態を表示する<br>販売可能ランプ：販売が可能であれば点灯する。<br>売り切れランプ：売り切れであれば点灯する。<br>釣銭切れランプ：釣銭切れであれば点灯する。<br>２．操作の入力を行う。入力は数字で行い、範囲外の入力は無いものとする。<br>0:10 円投入、1:50 円投入、2:100 円投入、3:500 円投入<br>4,5,6:　メニューのボタン、　7:ｷｬﾝｾﾙ、<br>３．操作の結果、発券、返金があれば返却口に出力する。なければ、１．より繰り返す。なお、返金は大きな硬貨より順に返金することとする。 |


©2011 uaubn